PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ^{6} : H04L 9/14, G06F 17/60	A1	(11) International Publication Number: WO 98/33296 (43) International Publication Date: 30 July 1998 (30.07.98)

(21) International Application Number: PCT/AU97/00889

(22) International Filing Date: 30 December 1997 (30.12.97)

(30) Priority Data:
PO 4749 23 January 1997 (23.01.97) AU

(71) Applicant *(for all designated States except US)*: COMMONWEALTH BANK OF AUSTRALIA [AU/AU]; 48 Martin Place, Sydney, NSW 1155 (AU).

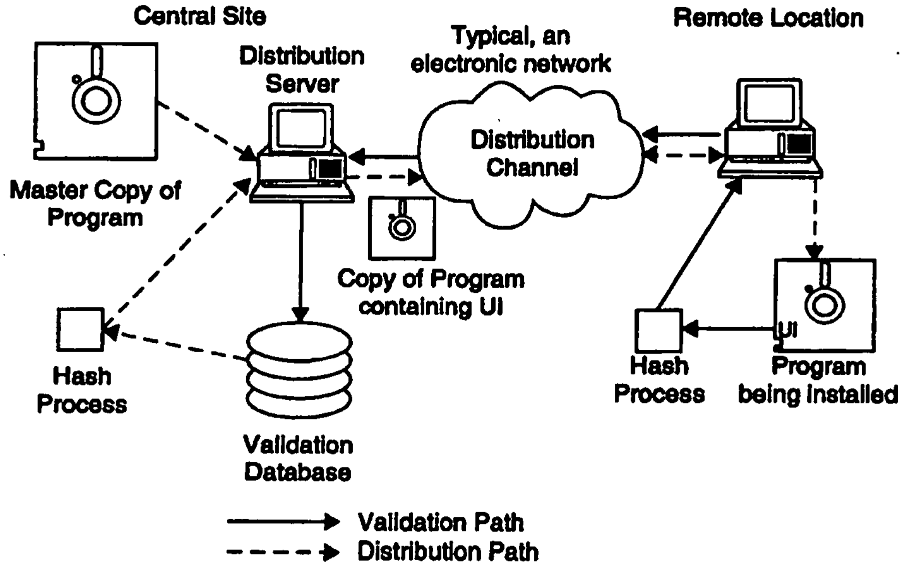
(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: MAPSON, Michael, Joseph [AU/AU]; 69 Yalor Road, Bangor, NSW 2234 (AU). COLLINS, Lyal, Sidney [AU/AU]; 1/37 Walton Crescent, Abbotsford, NSW 2046 (AU).

(74) Agent: WATERMARK PATENT & TRADEMARK ATTORNEYS; Level 4, Amory Gardens, 2 Cavill Avenue, Ashfield, NSW 2131 (AU).

(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, GM, GW, HU, ID, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, US, UZ, VN, YU, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG).

Published
With international search report.

(54) Title: DISTRIBUTION SYSTEM WITH AUTHENTICATION



(57) Abstract

The present invention relates to distribution systems, particularly those in which the delivery of goods/services can be authenticated as to their integrity or condition of delivery. In one particular, but not exclusive use, the present can be used to verify or authenticate the distribution and use of software via a relatively insecure environment. The present invention stems from the realisation that distribution verification and authentication can be provided by "attaching", associating and/or incorporating an Integrity Check Value (ICV) or some form of identification discernible only by the distributor and the distributed product to each product or service so distributed.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania
AM	Armenia
AT	Austria
AU	Australia
AZ	Azerbaijan
BA	Bosnia and Herzegovina
BB	Barbados
BE	Belgium
BF	Burkina Faso
BG	Bulgaria
BJ	Benin
BR	Brazil
BY	Belarus
CA	Canada
CF	Central African Republic
CG	Congo
CH	Switzerland
CI	Côte d'Ivoire
CM	Cameroon
CN	China
CU	Cuba
CZ	Czech Republic
DE	Germany
DK	Denmark
EE	Estonia
ES	Spain
FI	Finland
FR	France
GA	Gabon
GB	United Kingdom
GE	Georgia
GH	Ghana
GN	Guinea
GR	Greece
HU	Hungary
IE	Ireland
IL	Israel
IS	Iceland
IT	Italy
JP	Japan
KE	Kenya
KG	Kyrgyzstan
KP	Democratic People's Republic of Korea
KR	Republic of Korea
KZ	Kazakstan
LC	Saint Lucia
LI	Liechtenstein
LK	Sri Lanka
LR	Liberia
LS	Lesotho
LT	Lithuania
LU	Luxembourg
LV	Latvia
MC	Monaco
MD	Republic of Moldova
MG	Madagascar
MK	The former Yugoslav Republic of Macedonia
ML	Mali
MN	Mongolia
MR	Mauritania
MW	Malawi
MX	Mexico
NE	Niger
NL	Netherlands
NO	Norway
NZ	New Zealand
PL	Poland
PT	Portugal
R	Romania
RU	Russian Federation
SD	Sudan
SE	Sweden
SG	Singapore
SI	Slovenia
SK	Slovakia
SN	Senegal
SZ	Swaziland
TD	Chad
TG	Togo
TJ	Tajikistan
TM	Turkmenistan
TR	Turkey
TT	Trinidad and Tobago
UA	Ukraine
UG	Uganda
US	United States of America
UZ	Uzbekistan
VN	Viet Nam
YU	Yugoslavia
ZW	Zimbabwe

DISTRIBUTION SYSTEM WITH AUTHENTICATION

FIELD OF INVENTION

The present invention relates to distribution systems, particularly those in which the delivery of goods / services can be authenticated as to their integrity or condition of delivery. In one particular, but not exclusive use, the present can be used to verify or authenticate the distribution and use of software via a relatively insecure environment.

BACKGROUND

The distribution of software via insecure and untrusted channels is a risk factor when using open, public or untrusted network.

A complementary risk stems from ensuring that remotely operated software is operating in an unaltered manner, including all internal functions and internal data values.

These risks apply particularly to Internet commerce, but include all other arenas where the recipient of information sourced or processed remotely requires assurance that the information is protected in a known manner.

SUMMARY OF INVENTION

An object of the present invention is to provide a system, method and / or device which can check the integrity of software distributed over an at least partially insecure network.

To this end, the present invention provides a system for distributing goods and / or services over a medium which is at least partially insecure, the system including:

a means for establishing an Integrity Check Value (ICV),

storage means associated with the distribution of the goods and / or services, and

comparison means evaluating whether the goods and / or services after distribution have the same ICV as the goods and / or services before distribution.

The present invention also provides a method of distributing one or more copies of a goods and / or services based product, the method including the steps of:

determining a unique identification (UI) value for the product,

determining an ICV

encrypting the ICV

storing the ICV at a first location

recalculating the ICV in a manner determinable from both the first location and the product,

distributing a copy of the product to a second location remote from the first location, the distributed product having associated with it the recalculated ICV, and

comparing the ICV of the distributed product with the ICV known to the first location.

The encryption of the ICV may be based on the product and/or the Unique Identifier (UI).

Furthermore, the present invention provides a mechanism to distribute and use software in an insecure environment with relatively assured integrity and operation..

The present invention stems from the realisation that distribution verification and authentication can be provided by 'attaching', associating and / or incorporating an Integrity Check Value (ICV) or some form of identification discernible only by the distributor and the distributed product to each product or service so distributed.

In one form, the present invention enables software programs to be distributed to remote locations via any channels, including untrusted transmission networks, with the relative assurance that the received software has not been altered or modified in any manner.

The present invention describes a method and apparatus to distribute software programs via insecure distribution means with the ability to assure the integrity of these programs, both on first installation and all subsequent use of the program.

Programs may be distributed by electronic means in public, open or private networks, or by inclusion on storage media.

The present invention discloses a method and apparatus to verify that the program(s) have been installed in an unaltered state, or as in the intended manner of the program's creator, to a central (or trusted) server.

A method and apparatus is also specified to enable said programs to prove integrity to a central (or trusted) server, via untrusted networks, upon activation and at any subsequent time.

A consequence is that remotely located programs, possibly operating in untrusted environments, can be relatively assured they operate in the approved or desired manner, such approved manner may include ensuring that the program, or the user of the program, is communicating with the network entity (server) they intend to, while the network entity (server) can be relatively assured that the remote program operating in a known manner.

Each distributed copy of a program may be individually and uniquely identified.

Usage of individual copies of programs may be monitored.

The time, date and frequency of use of individual programs may be monitored for signs of abnormal or undesirable usage patterns.

The program may refuse to perform further operations unless integrity has been verified by the central (or trusted) server.

Individual copies of programs which operate to or through a central service server may be disabled remotely by setting a validity indicator at the central (or trusted) server to a "false" value.

A preferred embodiment of the present invention will now be described with reference to the accompanying drawings, in which:

Figure 1 illustrates one form of the distribution and initial validation,

Figure 2 illustrates one form of subsequent validation,

Figure 3 illustrates one form of Validation request,

Figure 4 illustrates one form of validation approval, and

Figure 5 illustrates in schematic form an apparatus for implementing the validation of the present invention.

Although the ensuing description deals with software, it should be appreciated that the present invention also has application to other goods / services, such as, but is not limited to, copyright material, security functions used for e.g. financial services over open or insecure n tworks and subscription services where consumption of services require verification and monitoring.

In this embodiment, assume that software programs are distributed from a central location, and contain a unique identification (not necessarily a serial number) (UI) value. This value can be embedded in the file or program, making the file or program a binary unique sequence.

By the use of cryptographic hashing techniques, each copy of the software therefore has a cryptographically unique hash value (Integrity Check Value or ICV).

A database containing the unique identification values and corresponding hash values can also be maintained at a central site.

As the specific copy of the software is activated, the hash value is re-calculated in a specific manner known to both the software and the central site.

The result of the received hash calculation / recalculation is electronically communicated to the central site, and validated against ICV entries in the database. If the validation is successful, the program can continue to operate as intended.

It is also possible, once a positive on-line validation has occurred, to extend the process, dependent on application requirements, to embed a new or derived Unique Identifier (UI). This could occur at initial distribution verification, or by updating the program at each "logon" to a central or validation server (VS). Subsequent validations would thus produce a "rolling" ICV, enabling unauthorised copies of a program to be identified should logon with an unrolled and therefore incorrect ICV occur

If, for example, the software has been corrupted or tampered with during or following its distribution or subsequent installation, the software will produce a result, within the limits of the chosen hashing algorithm, which is a different ICV to that known or expected by the central site, and hence the validation should fail. In that event, the use of that software can be terminated or the copy deleted or rendered otherwise unusable by a suitable means or command.

Commonly, this validation will occur during the initial installation of the software, and / or on subsequent use of the software. This ensures that software has a chain of authentication from initial installation and preferably through each use of the program.

Typically, the result of the calculated hash value will be included in every transaction, message or communication session generated by the remote software program. This may be used to provide an audit log of software usage, supplementing audit logs of user activity, which is uniquely linked to both the specific copy of the software program and every resulting transaction.

The operation of the present invention involves either one or two distinct phases.

These phases are the initial distribution of the software program with integrity, and validating the program integrity with ongoing use.

Download Integrity Mechanisms

Four exemplary techniques are described to ensure software integrity during the download/install phase. These are:

D1 Distribute the software with a unique identification value embedded by the distributor.

D2 Distribute the software as an identical copy of a "master version", and distribute the unique identification value by another channel, to be embedded at time of installation by a trusted tool distributed with the software.

D3 As with D1 above, with the addition of a challenge/response step.

D4 As with D2 above, with the addition of a challenge/response step.

D5 As with D3 or D4, with the inclusion of an offset pointer to be used in the ICV calculation process. The offset may be calculated in any deterministic manner which value could be a function of, or incorporate the UI value.

D6 As with D3 and D4 but with the inclusion of a direction indicator to indicate the direction of program data through the ICV calculation process, e.g. from start-of file to end of file or alternatively from end of file to start of file.

D7 As with D6 but with the inclusion of an offset pointer for use with the ICV calculation. The offset may be calculated in any deterministic value and be a function of, or incorporate the UI value.

An example of Challenge/Response, in this instance, might be a process whereby the Validation Server (VS), (or some VS agent which makes the data known to the VS), issues some, e.g. date/time dependent, random and recorded, data to the software requesting validation. The data could then be included in the ICV (integrity check value) calculation and (VAL_REQ) message, sent to the validation server. The server is capable of calculating the modified request ICV and whether it is returned within a valid time frame etc. If the VS challenge is responded to correctly within the set parameters, than a VAL_OK message may be issued.

Some Operational Integrity Mechanisms

Exemplary techniques might be, but are not confined to:

V1 The ICV or validation value is a simple hash of the executable file.

V2 The ICV is the result of hashing the executable file by use of a pre-calculated offset point in the file as the start of the input to the hash calculation process and continuing until the entire program has been used in calculating the ICV hash value. This offset pointer may be based upon the unique identification value, a fixed pointer chosen by the programmer, or determined in some other manner.

V3 As with V1, with the inclusion of an initialisation value. The initialisation value is included with the program file in the ICV calculation process. The initialisation value is originated by either the Validation Server or the program itself and communicated between the two locations to allow use in duplicate calculations at both locations.

V4 As with V2 above, with the inclusion of an initialisation value, originated by either the Validation Server or the program itself. The initialisation value is included with the file in the hash calculation process. The initialisation value is originated by either the Validation Server or the program itself and communicated between the two locations to allow use in duplicate calculations at both locations.

V5 A combination of V2 and V4. The offset pointer is calculated in a manner which includes the initialisation value. The initialisation value is included in the ICV calculation process. The initialisation value is originated by either the Validation Server or the program itself and communicated between the two locations to allow use in duplicate calculations at both locations.

V6 As with V2, with the addition of a direction indicator. The ICV is based upon the program file, an offset pointer and a direction flag. The ICV is the hash result from the direction that the program file is processed in (e.g. towards the End Of File or towards the Start Of File) and the offset pointer.

V7 As with V6, with the inclusion of a initialisation value. The initialisation value is originated by either the Validation Server or the program itself communicated between the two locations to allow use in duplicate calculations at both locations.

The hashing process is well known to those skilled in the art. The hashing process selected should be robust and fit for purpose. Example of hashing processes which might be employed are, (but not confined to), e.g. the NIST:-Secure Hash Standard, or the MD series of algorithms

Simplified Diagram of Distribution and Initial Validation Process

This description refers to figure 1 and presumes the Distribution Server performs both distribution and validation functions. This is not necessarily the case, since these two functions may be performed in separately or by separate systems.

The Distribution Server possess the Public and Private Key components of an Asymmetric Encryption algorithm. A Symmetric algorithm and process could be used, in addition to or as an alternative.

The Master Copy of the program is distributed with the unique identifier (UI) and the Public Key of the Distribution Server (PKDS). The UI may be embedded at time of distribution or embedded later by a trusted process.

The Distribution Server calculates the Integrity Check Value (ICV) from a copy (identical image) of the distributed Master and stores both the UI and ICV into the Validation Database.

When the remote location receives the program, the installation process occurs, which concludes with the received program being subjected to the same calculation process as used at the Distribution Server, to achieve a recalculated ICV.

The recalculated ICV and UI are sent to the Distribution Server, encrypted by PKDS in a Validation Request (VAL_REQ) message, see figure 3.

On receipt at the Distribution Server, a Private Key is used to recover the ICV and UI. The UI and received ICV components are then compared against the values calculated at time of distribution. By necessity, the ICV values are transmitted in encrypted form. The encryption process uses Public Key methods to protect the UI and ICV values from eavesdroppers. It is not necessary, in all cases, to transmit the Public Key, since this may be known within a closed network, or distributed by other means. However and alternatively, the VAL_OK message may be accompanied by a Public Key certificate. The most appropriate implementation should be determined by the application requirements or some affiliated standard.

If the stored and recalculated values match, the database entry for this specific copy of the program is updated to reflect the successful validation. Otherwise, the database entry is flagged as invalid. The program is also notified, by means of a Validation Successful (VAL_OK) message, see figure 4.

Simplified Diagram of Subsequent Validation Process

Referring to figure 2, when the remote location activates the program in response to some use action or automatic stimulus, the program may use the same calculation process used at the Verification Server, to achieve a recalculated ICV.

The recalculated ICV is sent to the Validation Server in a VAL_REQ message, where it is compared against the values calculated and validated, at the time of distribution. Again, the UI and ICV values are encrypted by Public Key methods, to prevent other parties correlating the two values, and creating software which may masquerade as valid software.

If the stored and recalculated values match, the program is notified of this fact by th us of a digitally signed VAL_OK (effectively a "ticket of authenticity") message which allows the program to proceed with processing information in accordance with the capabilities included by the program author. The digitally signed "Ticket of Authenticity" can in fact be validated by any entity possessing the Public Key of the Validation Server (PKVS). The digital signing of the Ticket of Authenticity is effected by using the protected and secret key of the validation server (SKVS), at that server. The technique, using asymmetric cryptography, is well known to those skilled in the art.

The Validation and Distribution Servers are shown as different systems, but in practice, it is likely that they will be different functions of the same system. If the Validation and Distribution Servers are different systems, then all distributed programs must contain appropriate public keys (i.e. PKDS and PKVS) to achieve the necessary secure communication paths required for software validation.

The program, once validated, may then communicate, with time correlated authenticity, to other systems in the electronic network, provided the digitally signed VAL_OK message is transmitted along with the request for connection on the remote network device.

Referring to figures 3 and 4, the Program Sequence Identifier (PSI) assists in preventing replayed validation request messages.

The Validation Server Sequence Identifier (VSSI) further assists in preventing replayed validation request messages.

The Date/Time stamp (DT) enables any recipient to determine when the program was last validated. The recipient of a message or transaction containing a VAL_OK component may then make a valued decision on the validity of the software generating the message, and therefore the reliability which may be placed on the specified transactional data within the message. The recipient of such a message may also check the authenticity of the VAL_OK components by reference to the digital signature on the VAL_OK or by reference to the Validation Server.

An initialisation vector is, referring to Figures 3 and 4, a value to be common to both the software or device under registration and the registration entity (validation server). The value is used in the process of calculating the ICV and therefore preferably should occur on a "one time only" basis. This will ensure that each validation is unique in content and cannot be replayed etc. An initialisation vector value might be derived from, (but is not confined to), e.g. the serial number of the device or software, a transaction counter or a date/time stamp. The nett result being a value used to derive a one time token of authenticity for the ICV calculation.

Referring to Figure 5, a schematic illustrates a means of implementing the present invention. In the figure, reference numerals denote:

1. **The Software Module:** The goods/services or program being assured through this process.
2. **ICV Calculation process:** The process that calculates the ICV value, using a Hash Function and other optional processing steps, which may include use of an Initialisation Value or other token, an offset pointed, and direction flag.
3. **Direction Flag or indicator:** Which may be used to indicate the direction to process the Software Module in calculating the ICV. The Direction Flag, or indicator, will indicate to process from Start of File to End of File, or from End of File towards Start of File. Typically, this will be determined from the UI and optionally, other values such as IV.
4. **Initialisation Value:** An instance or program specific value which may be used in calculating the ICV.
5. **Offset pointer:** A value which may be used as to indicate an offset start point for processing the Software module as it is processed for ICV calculations. Typically, this will be determined from the UI and optionally, other values such as IV.

THE CLAIMS DEFINING THE INVENTION ARE AS FOLLOWS:

1. A system for distributing goods and / or services over a medium which is at least partially insecure, the system including:

a means for establishing an Integrity Check Value (ICV),

storage means associated with the distribution of the goods and / or services, and

comparison means evaluating whether the goods and / or services after distribution have the same ICV as the goods and / or services before distribution.

2. A system as claimed in claim 1, in which the established ICV is stored in the storage means, and is incorporated or attached to the distributed goods and / or service.

3. A system as claimed in claim 1 or 2, in which the goods and / or services are software based.

4. A system as claimed in claim 3, in which the comparison means evaluates the ICV at a time proximate installation of the software.

5. A system as claimed in claim 3 or 4, in which the comparison means evaluates the ICV at or during use of the software.

6. A system as claimed in any one of claims 1 to 5, in which the ICV in which ICV is established based on the UI and the program data.

7. A method of distributing one or more copies of a goods and / or services based product, the method including the steps of:

determining a unique identification (UI) value for the product,

encrypting, calculating and encrypting the ICV, based on the product and the Unique Identifier (UI), storing the ICV at a first location

recalculating the ICV in a manner determinable from both the first location and the product,

distributing a copy of the product to a second location remote from the first location, the distributed product having associated with it the recalculated ICV, and

comparing the ICV of the distributed product with the ICV known to the first location.

8. A device adapted to perform the method as claimed in claim 7.

9. A method, apparatus, or system as herein disclosed.

Fig.1

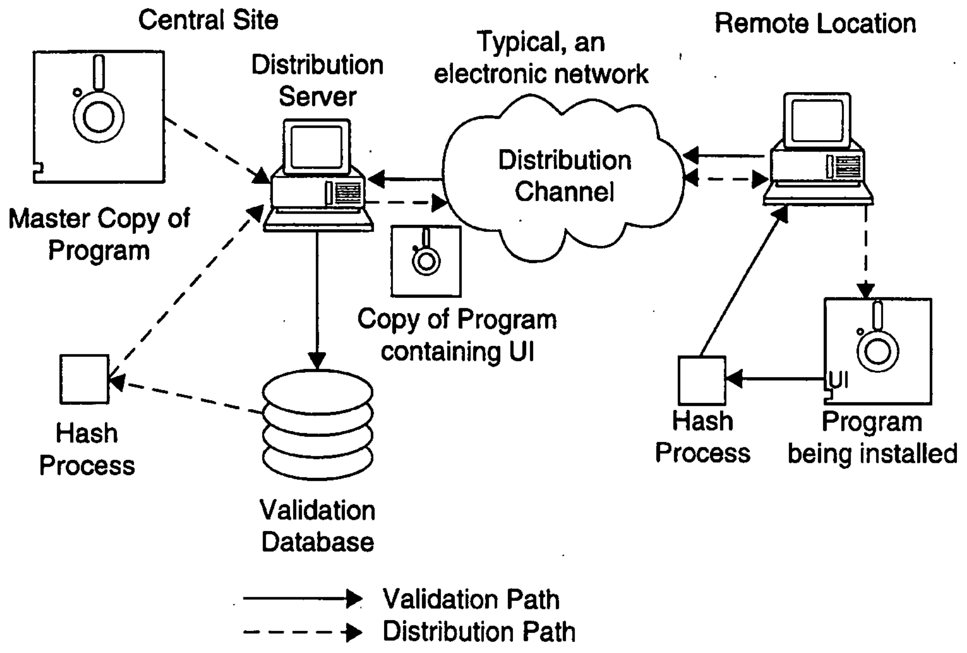

Central Site
Distribution Server
Typical, an electronic network
Remote Location
Distribution Channel
Master Copy of Program
Copy of Program containing UI
Hash Process
Validation Database
UI
Hash Process
Program being installed
Validation Path
Distribution Path

Fig.2

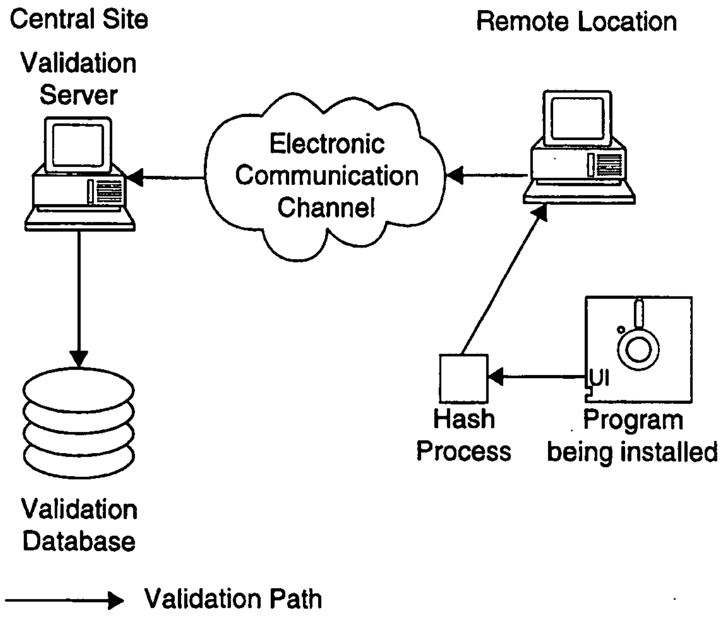
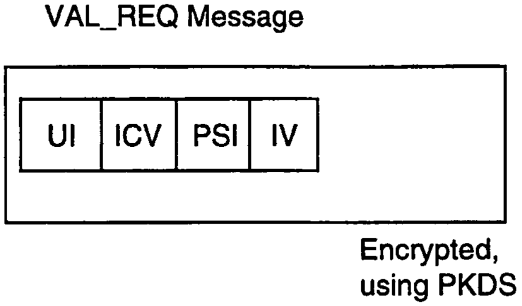

Central Site
Validation Server
Remote Location
Electronic Communication Channel
Validation Database
UI
Hash Process
Program being installed
Validation Path

Fig. 3

VAL_REQ message



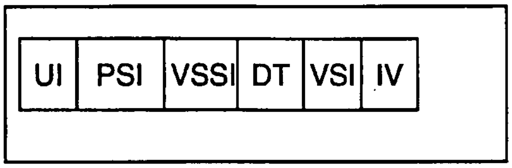
UI	Unique Identifier of Program
PSI	Program Sequence Identifier
VSSI	Validiation Server Sequence Identifier
DT	Date and Time of this Validation
VSI	Validation Server Identifier
IV	Initialisation Vector

The Program Sequence Identifier assists in preventing replayed validation request messages.

Fig. 4

VAL_OK message

VAL_OK Message



Encrypted,
or Digitally Signed,
by SKVS

UI	Unique Identifier of Program
PSI	Program Sequence Identifier
VSSI	Validiation Server Sequence Identifier
DT	Date and Time of this Validation
VSI	Validation Server Identifier
IV	Initialisation Vector

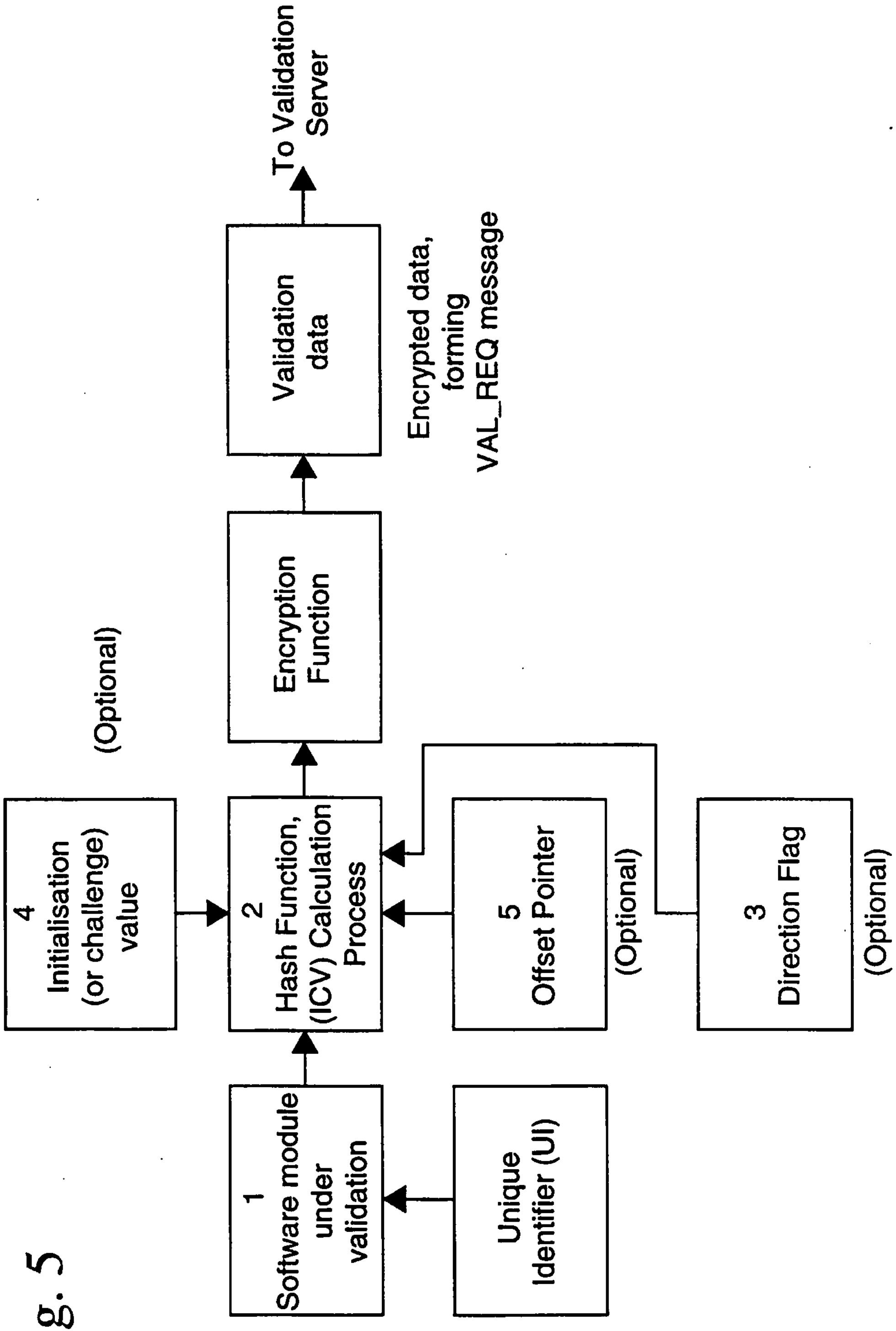

1
Software module under validation
Unique Identifier (UI)
2
Hash Function, (ICV) Calculation Process
4
Initialisation (or challenge) value
(Optional)
5
Offset Pointer
(Optional)
3
Direction Flag
(Optional)
Encryption Function
Validation data
Encrypted data, forming VAL_REQ message
To Validation Server

Fig. 5

INTERNATIONAL SEARCH REPORT

International Application No.
PCT/AU 97/00889

A. CLASSIFICATION OF SUBJECT MATTER

Int Cl^{6}: H04L 9/14; G06F 17/60

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC^{6}: as above

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
AU: IPC as above

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
WPAT, INSPEC: (distribution, secur:, software)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP 686906 (SUN MICROSYSTEMS) 13 December 1995 Abstract, columns 4, 5, figures	1-9
X	"Secure distribution of electronic documents in a hostile environment" (Rubin, A D) Computer Communications Volume 18 No: 6 June 1995 pages 429-434	1-5
Y	Whole document	6-9
X	"To Whom am I speaking?" (LOMAS et al) IEEE Computer Volume 28 No: 1, January 1995 pages 50-54 especially Solution Strategy and figures	1-9

☒ Further documents are listed in the continuation of Box C

☒ See patent family annex

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search 4 February 1997	Date of mailing of the international search report 17 FEB 1998
Name and mailing address of the ISA/AU AUSTRALIAN INDUSTRIAL PROPERTY ORGANISATION PO BOX 200 WODEN ACT 2606 AUSTRALIA Facsimile No.: (02) 6285 3929	Authorized officer DALE E. SIVER Telephone No.: (02) 6283 2196

INTERNATIONAL SEARCH REPORT

International Application No.
PCT/AU 97/00889

C (Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	"Secure Software Distribution" (ROSENBLIT) IEEE Network Operations and Management Symposium Volume 2, 14-18 February 1994 pages 486-496 Whole document	1-9
Y	WO 92/09160 (TAU SYSTEMS CORP) 29 May 1992 Abstract, figures	1-9
A	"Location-Independent Naming for Virtual Distributed Software Repositories" (BROWNE et al) ACM special issue August 1995 pages 179-185 especially section 5 on Authenticity, Integrity and Consistency of Resources	1, 7
A	WO 94/16508 (INFONOW CORP) 21 July 1994 Abstract, figures	1, 7

INTERNATIONAL SEARCH REPORT
Information on patent family members

International Application No.
PCT/AU 97/00889

This Annex lists the known "A" publication level patent family members relating to the patent documents cited in the above-mentioned international search report. The Australian Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

Patent Document Cited in Search Report		Patent Family Member					
EP	686906	JP	8166879				
WO	92/09160	CA	2095723	EP	556305	JP	6501120
		US	5103476	US	5222134		
WO	94/16508	AU	59906/94				

END OF ANNEX